MÄK RTI Release 1.3.1 Release Notes

The MÄK RTI (Run-Time Infrastructure) 1.3.1 is an implementation of a subset of the HLA Interface Specification, version 1.3. It implements the same API (exact same header files) as the DMSO RTI, version 1.3v4, 1.3v5, and 1.3v6.

This document covers the following:

System Requirements

Table 1 lists the platforms supported by the MÄK RTI 1.3.1. Application code must be built with the indicated compilers in order to interface to the RTI libraries.

Table 1: Platforms supported

Platform	Compiler
SGI IRIX6.2 or later (both n32 and o32 ABIs)	MipsPro7.1 C++ compiler (available from SGI)
Sun Sparcstation with SunOS4.1.3 or later	GNU C++ (g++) version 2.7.2.3 and G++ libraries version 2.7.2. (Available via anonymous FTP from prep.ai.mit.edu/pub/gnu.)
Sun Sparcstation with Solaris2.5 or later	SPARCompiler C++ version 4.1 (available from Sun)
PC with Red Hat Linux 5.1	GNU C++ (g++) distributed with Red Hat distribution. The command: `g++ -v` returns: `gcc version egcs-2.90.27 980315 (ecgs-1.02 release)`
PC with Windows NT or Windows 95/98	Microsoft Visual C++ 5.0 or later.

Compatibility with the DMSO RTI

With the exception of the one additional function (RtReadFileDescriptor, described in MÄK RTI documentation), the MÄK RTI has the exact same API as the DMSO RTI version 1.3v4, 1.3v5 and 1.3v6. Unless you are using this optional function, an application built against the DMSO RTI does not need to be recompiled in order to be used with the MÄK RTI, or vice versa.

The MÄK RTI 1.3.1 uses the same FED file format as DMSO RTI 1.3v4, 1.3v5 or 1.3v6, but a different RID file format. (RIDs are RTI implementation-dependent.)

Network Compatibility

The MÄK RTI is not network compatible with the DMSO RTI. (In general, no two RTI implementations will be network compatible due to differences in the low-level network mechanisms that they use (which include, but are not limited to packet layout). Applications that want to interoperate in the same federation execution must all be using the same RTI implementation. A federation can all easily switch from one RTI implementation to another between runs, but during each run, all participants must agree on which RTI to use, much as they must also agree on which FOM to use.

Backward Compatibility

Applications that were built again the MÄK RTI 1.3 will work with the MÄK RTI 1.3.1 without recompiling. MÄK RTI 1.3.1 is network compatible with MÄK RTI 1.3 unless you are using features new to the MÄK RTI 1.3.1.

Documentation

Because the MÄK RTI implements the same interface specification as the DMSO RTI, the documentation provided with the DMSO RTI applies to the MÄK RTI as well, apart from the few differences we describe in this release note and the MÄK RTI Reference Manual. The DMSO RTI documentation is included with the MÄK RTI in the *doc* directory. It is also available on DMSO's HLA web site: http://hla.dmso.mil. Choose HLA Software Distribution Center, then click on the Products button.

Services Implemented

MÄK RTI 1.3.1 implements a subset of the RTI services - the functionality that is most likely to be needed by the real-time platform-level simulation community.

In addition to the services supported in MÄK RTI 1.3, the MÄK RTI supports the following federate-initiated services:

- RegisterFederationSynchronizationPoint
- SynchronizationPointAchieve

and the following RTI-initiated services:

- AnnounceSynchronizationPoint
- FederationSynchronized
- SynchronizationPointRegistrationFailed
- SynchronizationPointRegistrationSucceeded

MÄK RTI 1.3.1 does not publish any MOM data.

Federation Time

The MÄK RTI 1.3.1 files *libfedtime.so* or *libfedtime.dll* implement the same subclass of RTI::FedTime as DMSO's file *libfedtime*. This subclass is called *RTIfedTime* and its definition is in *fedtime.h*. You may use either the MÄK RTI's *libfedtime* or the DMSO RTI's *libfedtime* with either MÄK or DMSO's *libRTI*. (In MÄK RTI 1.3, our *libfedtime* implemented a different subclass of RTI::FedTime called *DtVrlFedTime*.)

New Features

MÄK RTI 1.3.1 has the following new features:

- New RID configuration parameters
- Support for synchronization points
- Support for reliable communication
- The method of setting up and configuring the FLEXlm license server has been simplified.

The RID File

MÄK RTI 1.3.1 has three new RID configuration parameters:

- RTI_internalMsgReliable - Indicates whether or not you want to send the RTI's internal bookkeeping messages using reliable transport. Values are:
  - 0 - Use best-effort for all internal messages
  - 1 - Use reliable for all internal messages
- RTI_fomDataReliable - Indicates whether or not you want to send object attributes and interactions using reliable transport when so stated in your FED file. Values are:
  - 0 - Use best-effort for all FOM data.
  - 1 - Use reliable or best-effort as indicated in the FED file.
- RTI_tcpForwarderAddr - Contains the IP Address of the machine that the rtiexec application is running on. This value is required if using any type of reliable transport.

Example:

```
(setqb RTI_internalMsgReliable 1)
(setqb RTI_tcpForwarderAddr "123.145.167.189")
```

Synchronization Points

To use the synchronization points services, you must be using the rtiexec. To specify use of the rtiexec, in *rid.mtl* file, set the **RTI_useRtiexec** parameter to 1.

Reliable Transport

There are multiple ways to use reliable transport. If you want the RTI's internal bookkeeping messages to be send using reliable transport, such as federate subscription, then you need to set the *rid.mtl* file parameter **RTI_internalMsgReliable** to 1.

Additionally, you can indicate in your FED file which object attributes and interactions you want to be sent by reliable transport or best effort. If you want this data sent as stated in your FED file, then you must set the *rid.mtl* file parameter **RTI_fomDataReliable** to 1. Otherwise, all FOM data will be sent best-effort.

You must run the rtiexec application if you are using reliable transport for either internal messages or FOM data. When using reliable transport, set the *rid.mtl* file parameter **RTI_useRtiExec** to 1. In addition, you must set the *rid.mtl* file parameter **RTI_tcpForwarderAddr** to the IP address of the machine that is running the rtiexec application.

For example if you want to be using reliable transport for your internal messages and FOM data and the machine that you are running the rtiexec on has an IP address of 123.123.123.123, then your *rid.mtl* file should have the following:

```
(setqb RTI_useRtiExec 1)
(setqb RTI_internalMsgReliable 1)
(setqb RTI_fomDataReliable 1)
(setqb RTI_tcpForwarderAddr "123.123.123.123")
```

The rtiexec application creates a TCP Forwarder that manages the receiving of messages from individual federates and is responsible for forwarding the messages to all the other federates in the federation.

The *rid.mtl* file needs to be in the same directory as your FED file - either the directory indicated by the RTI_CONFIG environment variable, or the current directory if RTI_CONFIG is not set.

Minor Changes

- The standard math library (*libm*) is now linked into *libRTI*, so that applications do not have to be linked with *libm*.
- enableAttributeScopeAdvisory service is still not implemented, but it is now a no-op, and does not throw an RTI::RTIinternalError exception.

Changes to FLEXlm Setup and Configuration

With this release of the MÄK RTI, we have simplified the process for configuring the FLEXlm license management. It is no longer necessary to put a license file on each client machine, and you no longer have to merge license files when you purchase new products or additional licenses.

If you do not have a prior installation of MÄK Technologies products, follow the instructions in "Setting Up and Using the FLEXlm License Manager" on page 8. If you already have MÄK products installed, your upgrade procedure depends on the version of FLEXlm that you are running. Table 2 lists MÄK products and how to proceed.

Table 2: Product versions and license management

Product	FLEXlm Version	Procedure to Follow
VR-Link 3.0, 3.1 Stealth for SGI 3.1, 3.1.1 Stealth for PCs 3.2.1 Logger 3.1	5.x	Follow installation procedure in original product documentation. Merge license files as described in "Merging New License Files for Products Using FLEXlm 5.x" on page 7.
VR-Link 3.2 or later Stealth for SGI 3.2 or later Stealth for PCs 3.3 or later Logger 3.2 or later Plan View Display 1.x	6.0	Follow the instructions in "Updating Your Existing License Management Configuration" on page 6.

Updating Your Existing License Management Configuration

If you have FLEXlm license management configured for products you purchased prior to the current release, you must do the following to update your license management configuration:

1. On the license server machine, rename the *license.dat* file to *mak.lic* (or any name ending in *.lic*). (If you are unsure about the distinction between the license server machine and client machines, see "The FLEXlm Client-Server Architecture" on page 8.)
2. Open the license file in a text editor.
3. Remove the port number from the SERVER line. For example, if the first line of the license file is:

   ```
   SERVER oak 0800359de785 4567
   ```

   the port number is 4567. Delete it from the file.
4. Remove the LM_LICENSE_FILE environment variable from the server machine and all client machines unless another vendor requires it.

5. Delete the *license.dat* file from all client machines.
6. Set the MAKLMGRD_LICENSE_FILE environment variable on all client machines (including the license server machine if you run MÄK Technologies products on it).

   The syntax for the environment variable is: *@Server_name*. For example, if the server machine is oak, set the environment variable to `@oak`.

Merging New License Files for Products Using FLEXlm 5.x

If you are using FLEXlm 5.x, you cannot use the new method of license management. If you receive a new license file from MÄK, you need to merge it with the old file.

1. Your new license file will look something like the following:

```
SERVER mollusc 006097752f93
DAEMON maklmgrd .
PACKAGE rti maklmgrd 1999.177 60D00041FA200D8A795D \
      COMPONENTS=rti1
INCREMENT rti maklmgrd 1999.177 1-jan-0 1 0BB480511615C175B8CF \
      PLATFORMS=i86_n
```

   Your existing license.dat file is similar to the following:

```
SERVER mollusc 006097752f93 4567
DAEMON maklmgrd “c:\Program Files\MÄK Technologies\stealth\flexlm\
      maklmgrd”
PACKAGE makproduct maklmgrd 1999.177 40C020D12D609AB04D55 \
      COMPONENTS="hprod1 dprod1 prod1"
INCREMENT makproduct maklmgrd 1999.177 1-jan-0 1 AB4450916DD1944435D2 \
      PLATFORMS=i86_n
```

Notes

- The server name and host id in the two license files should be identical. The new file will not have a port number.
- The line continuation marks (\) in the examples are for clarity of presentation. They may not be present in the files you receive.

2. Open up the two files in a text editor. Append everything except the Server and Daemon lines (the Server lines will be identical except for the port) from the new *license.dat* file to the end of the old *license.dat* file. For example:

```
SERVER mollusc 006097752f93 4567
DAEMON maklmgrd “c:\Program Files\MÄK Technologies\stealth\flexlm\
      maklmgrd”
PACKAGE makproduct maklmgrd 1999.177 40C020D12D609AB04D55 \
      COMPONENTS="hprod1 dprod1 prod1"
INCREMENT makproduct maklmgrd 1999.177 1-jan-0 1 AB4450916DD1944435D2 \
      PLATFORMS=i86_n
PACKAGE rti maklmgrd 1999.177 60D00041FA200D8A795D \
      COMPONENTS=rti1
INCREMENT rti maklmgrd 1999.177 1-jan-0 1 0BB480511615C175B8CF \
      PLATFORMS=i86_n
```

3. Put the updated license.dat file in the flexlm directory on the license server and on all machines on which you have the RTI installed.

" ' ' " #$% # "

The RTI, like other MÄK Technologies products, uses the FLEXlm license manager. Before you can use the RTI, you must obtain a valid license file and configure the license server and client machines. (For information about FLEXlm, view the FAQ page at www.globetrotter.com/flmfaq.htm.) Contact the MÄK Technologies sales department for information about special licensing agreements.

%#' * + , "

This section explains the FLEXlm architecture to provide a context for the setup instructions that follow. FLEXlm uses a client-server architecture. The FLEXlm executable, maklmgrd, runs on one computer, known as the license server. This machine services every machine (including possibly itself) on which you want to run a MÄK Technologies product. The machines on which the MÄK Technologies products are running are called clients.

The license server has a license file (supplied by MÄK Technologies). This file identifies the MÄK Technologies products for which you have licenses. On each client machine, the MAKLMGRD_LICENSE_FILE environment variable (which you must set) points to the server.

Licenses “float”, so you can install product software on more machines than you have licenses for, but at any given time, you can run only as many instances of a product as you have licensed.

The license key in the license file is tied to the host id of the server machine. Therefore, if you decide to change the server, you need to get a new license file. Then, you must update the MAKLMGRD_LICENSE_FILE environment variable on every client. This is not a trivial process, and is not recommended.

Figure 1 illustrates the client-server architecture for FLEXlm and MÄK products.

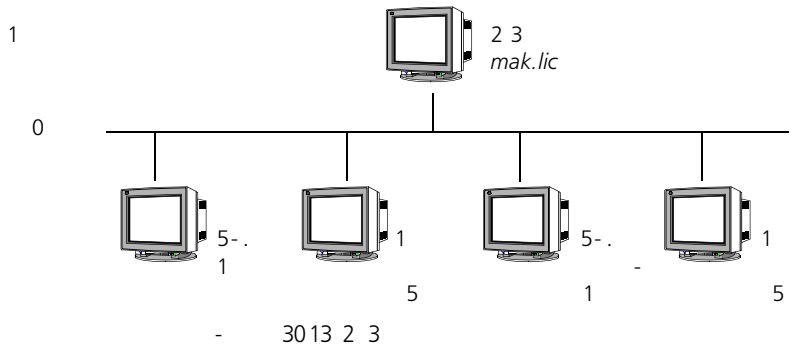

Figure 1. FLEXlm example architecture

Installing and Configuring the License Manager

The general procedure for installing and configuring license management is as follows (each step is explained in the rest of this section):

1. Install license manager software.
2. Request a license file.
3. Put the license file in the *flexlm* directory on the license server.
4. Set the MAKLMGRD_LICENSE_FILE environment variable on all client machines.

If you have MÄK Technologies products installed and you purchase additional licenses or products, you must get a license file for the new products or licenses. For this procedure, see "Adding Additional License Files" on page 12.

Installing the License Manager Software

As part of the installation of all MÄK products, the FLEXlm software is installed in the *flexlm* directory under the root product directory.

1. Install your MÄK products on the machine(s) on which you want to run them.
2. If this is a networked system, and you did not install MÄK software on the license server, copy the *flexlm* directory from a client machine to the license server machine.

Requesting A License File

License files are keyed to the host id of the license server. You need to supply MÄK with some information so that we can generate a license file for you.

To get a license file:

1. On the license server, in a command window, change to the *flexlm* directory.
2. Execute *lmhostid.* This program generates a unique number that MÄK uses to generate your license activation codes. Write down the number.
3. Go to http://www.mak.com/support/keyintro.htm and complete the license key request form. MÄK will e-mail you a file named *mak.lic*.

To get a license file, you need to know the name of the license server machine. To get the name of your machine follow the procedure for your operating system:

UNIX

At a command prompt, type `hostname`.

Windows 95/98/NT

1. On the Start menu, choose Settings → Control Panel.
2. Open the Network applet.
3. Select the Identification tab. The machine name is listed in the Computer Name field.

Putting the License File in the *flexlm* Directory

After you receive the license file, put it in the *flexlm* directory on the license server.

! If you already have a *mak.lic* file in the *flexlm* directory, do not overwrite the file. See the instructions in "Adding Additional License Files" on page 12.

Setting the MAKLMGRD_LICENSE_FILE Environment Variable

The MAKLMGRD_LICENSE_FILE environment variable identifies the server machine so that the FLEXlm software on a client can check out a license when you run a MÄK Technologies product. You must set MAKLMGRD_LICENSE_FILE on every client machine.

If you are running MÄK Technologies products on the license server machine, it is also a client, so you must set MAKLMGRD_LICENSE_FILE.

The syntax for the environment variable is: *@Server_name*. For example, if the server machine is oak, set the environment variable to `@oak`.

The following sections explain how to set environment variables on the different platforms that MÄK Technologies products run on.

Windows 95/98

On Windows 95/98, you set environment variables by adding or editing lines in the *Autoexec.bat* file. To set the MAKLMGRD_LICENSE_FILE variable, add a line similar to the following:

```
set MAKLMGRD_LICENSE_FILE=@oak
```


Do not put spaces around the equal (=) sign.

Windows NT

On Windows NT, you set environment variables in the Environment tab of the Properties dialog box for the machine:

1. Right-click the My Computer icon on the Desktop.
2. On the pop-up menu, select Properties. The System Properties dialog box opens (Figure 2).
3. Select the Environment tab.
4. In the Variable text field, type MAKLMGRD_LICENSE_FILE.
5. In the Value field, type the value, for example:

   `@oak`

Figure 2. Setting an environment variable in Windows NT

UNIX

On UNIX, you set environment variables in your *.cshrc* (or equivalent startup file). Set the variable similarly to the following example:

```
setenv MAKLMGRD_LICENSE_FILE @oak
```

Adding Additional License Files

If you buy additional MÄK products, or additional licenses for products you already have, you must get an additional license file.

> To request an additional license file, follow the same procedure as for your first license file. See "Requesting A License File" on page 10.

To add a new license file to your license server:

1. When you get the new license file, rename it from *mak.lic* to *mak2.lic*, or some other name ending in *.lic*, so that it has a different name from the license file(s) currently in the *flexlm* directory on the license server.
2. Put the renamed file in the *flexlm* directory on the license server, with the other license files.

Bug Fixes

- Solaris release does not crash when RTI_CONFIG is not set, and there is no *./rid.mtl*.
- RTI::Exception's copy constructor copies the _name field, so that its value is available when catching exceptions by value (not by reference).
- Constructing an attribute handle set with an argument of exactly MAX_ATTRIBUTES_PER_CLASS no longer causes an ArrayIndexOutOf-Bounds exception to be thrown.
- RTI::AttributeHandleValuePairSet's and RTI::ParameterHandleValuePairSet's moveFrom functions now copy data from one set to another, rather than moving it.
- Several memory leaks were fixed.